

Ordering Is Not Resolution: What the Instruction Hierarchy Defines, What It Leaves Undefined, and the Same-Tier Conflicts Agent Benchmarks Do Not Separate

Pranay M. Mahendrakar

SONYTECH

ORCID: 0009-0003-7224-029X

Life of Research

2026-08-28

Abstract

When a language-model agent receives instructions that conflict, the dominant remedy is a privilege ordering over sources: system above developer, developer above user, user above tool output. This paper argues that the ordering paradigm is well-defined for one class of conflict and undefined for another, and that published benchmarks and training sets do not separate the two. A conflict between instructions carrying different privilege labels has an answer the paradigm can state; a conflict between two instructions carrying the same label does not, because an ordering over privilege levels does not induce an ordering among the instructions inside a level. The second class is not hypothetical. One profiler of real deployed prompt policies reports that across thirteen thousand jointly governed trials, only about a third of cases satisfy both of two individually reasonable standing rules. Multi-principal deployments, where two users hold equal authority, instantiate the same structure by construction. The paper distinguishes three failure classes that the single phrase "instruction hierarchy failure" currently covers, argues that reported hierarchy-compliance numbers are sums over classes with different remedies, and states what a same-tier resolution rule would have to supply that an ordering does not. The case for the ordering paradigm is presented first and is not weak: several recent results report large, transferable gains from training on ordering. Nine studies that would settle the open parts are named. No experiments are reported here, and the strongest case against this paper's own position is stated in full.

1. Introduction

Language-model agents take instructions from more sources than a chat model does. A deployed agent reads a system prompt written by a platform, a developer prompt written by an application author, a user turn, the output of tools it calls, documents it retrieves, and messages from other agents. Wallace et al. (2024) identified the structural vulnerability this creates: models "often consider system prompts (e.g., text from an application developer) to be the same priority as text from untrusted users and third parties," which is what makes prompt injection work. The attack literature had already demonstrated the consequences, directly (Perez and Ribeiro, 2022) and indirectly through retrieved content (Greshake et al., 2023), and later at agentic scale with tool-calling environments (Debenedetti et al., 2024) and explicitly malicious task suites (Andriushchenko et al., 2025).

The remedy Wallace et al. (2024) proposed, and which the field adopted, is an instruction hierarchy: an explicit priority ordering over instruction sources, with training data that teaches a model to ignore lower-privileged instructions when they conflict with higher-privileged ones. The paradigm now has benchmarks (Zhang et al., 2025; McCauley et al., 2026), training sets (Guo et al., 2026; Zheng et al., 2026), diagnostic frameworks (Kariyappa and Suh, 2026), and proposed extensions to many levels (Zhang et al., 2026) and to solver-guided constraint satisfaction (Yang et al., 2026b).

This paper is about a structural property of that remedy. An ordering over a set of privilege levels answers the question "which of these two instructions wins" exactly when the two instructions sit at different levels. It says nothing when they sit at the same level, because an ordering over privilege levels does not induce an ordering among the instructions inside a level. This is not a claim that the paradigm is badly implemented. It is a claim about what the paradigm is defined on.

The distinction would be academic if same-level conflict were rare. It is not. Yan et al. (2026) profiled six public prompt policies, extracted 276 source rules, and found that of 13,335 jointly governed trials realised from hard-collision rule pairs, only 35.4 percent satisfied both governed rules. Those collisions are between standing rules inside one policy document, written by one author, carrying one privilege label. He et al. (2025) study conflicts among constraints inside a single user instruction, and report that models detect such conflicts well while rarely surfacing them. Yang et al. (2026a) formalise multi-user agents as a multi-principal decision problem in which one agent serves several users with distinct roles and conflicting interests, a setting where equal-authority conflict is the design, not an edge case.

The concern this paper raises is not that these failures are unsolved. It is that they are being counted, and reported, inside an aggregate whose name asserts a diagnosis. When a benchmark reports that a model achieves some percentage of "instruction hierarchy compliance," that number sums over conflicts an ordering can adjudicate and conflicts it cannot, and the two have different remedies. Three papers from inside the paradigm report symptoms consistent with this conflation. Guo et al. (2026) state that "IH failures can be confounded with instruction-following failures" and that models "can learn shortcuts such as overrefusing." Kariyappa and Suh (2026) decompose non-compliance into three distinct loci and report that which one dominates varies across models, tasks and context length. McCauley et al. (2026), evaluating 37 models, conclude that hierarchy robustness "is not a single capability, but a set of behaviors."

Sections 2 through 4 state the scope, the paradigm's inheritance from operating systems, and the case in its favour, which is stronger than the framing above may suggest. Sections 5 and 6 set out the three failure classes and the measurement problem. Sections 7 and 8 state what a same-tier rule would have to supply and name nine studies that would settle the open parts. Sections 9 and 10 argue against this paper's own position and disclose the quality of the evidence it rests on.

2. Scope, and what this paper does not claim

This paper is literature analysis. No experiments were run for it, no model was evaluated, and every number in it is attributed to the paper that measured it. The genre is analysis rather than survey: the aim is not to catalogue the instruction-hierarchy literature but to argue one point about its structure. Coverage is therefore deliberately partial, and the omissions are stated here rather than discovered later.

Four claims are explicitly not made.

First, this paper does not claim the instruction hierarchy is a bad idea, or that it fails at what it was designed for. Section 4 argues the opposite at some length, and the strongest recent results in the area are gains produced by ordering-based training.

Second, it does not claim that same-tier conflict is the largest source of agent failure. Nobody has measured the relative frequency of same-tier and cross-tier conflict in deployed traffic. That quantity appears in no source found during this search, and the argument here does not depend on it. What the argument depends on is that same-tier conflict is common enough to be measured at all, which Yan et al. (2026), He et al. (2025) and Yang et al. (2026a) establish.

Third, it does not claim that the field is unaware of same-tier conflict. Yang et al. (2026b) say directly

that prior instruction-hierarchy work "mainly focuses on adversarial attacks and overlooks the benign but common instruction conflicts that arise in real-world applications," and propose a method for partially and implicitly conflicting instructions. That paper is the closest existing answer to the problem named here, and Section 7 positions against it rather than around it.

Fourth, this paper does not propose a resolution mechanism. Section 7 states requirements a mechanism would have to meet. Requirements are cheaper than mechanisms, and a reader should discount the contribution accordingly.

The scope is also restricted to instructions, meaning imperative content that an agent may comply with or violate. Conflicts of belief, of retrieved factual content, or of memory are a different literature with a different structure and are out of scope here.

3. What the ordering paradigm inherited

Privilege ordering did not originate with language models, and the older literature is worth reading because it already knows the shape of the problem.

Schroeder and Saltzer (1972) described a hardware architecture for protection rings: a totally ordered set of privilege levels, with access rights that increase monotonically toward the centre. Saltzer and Schroeder (1975) set out the design principles that accompany such systems, including least privilege and complete mediation. The instruction hierarchy is recognisably a descendant. It posits a small totally ordered set of levels, assigns each instruction source to one, and resolves conflict by taking the maximum.

What rings do well is exactly what the instruction hierarchy does well: they prevent a less privileged principal from overriding a more privileged one. What rings do not do is adjudicate between two requests at the same ring. That was never their job; other machinery handled it.

The nature of that other machinery is the instructive part. Operating systems did not resolve same-ring contention by refining the ring structure until every pair of processes sat at different levels. They introduced mechanisms of a different kind: schedulers, locks, transactions, and arbitration protocols. These are not orderings. A scheduler is a procedure that may treat equals equally, take turns, or make a policy choice that the privilege structure does not contain. A transaction can abort and retry rather than pick a winner. The general lesson is that same-level contention was met with a new mechanism class, not with a finer partition of the old one.

This matters for reading the agent literature, because the dominant response to observed hierarchy failures has been to refine the ordering: more levels (Zhang et al., 2026), better training on the ordering (Guo et al., 2026), and better reasoning about the ordering (Zheng et al., 2026). Those are improvements to a mechanism of the first kind. Whether a mechanism of the second kind is needed is the question this paper is raising, and the systems history supplies a precedent rather than an answer.

The systems-management literature met the same limit when it moved from access control to policy. Moffett and Sloman (1993) developed policy hierarchies for distributed systems management, and Moffett and Sloman (1994) analysed policy conflict directly. Lupu and Sloman (1999) treated conflicts in policy-based distributed systems management at length, developing precedence relationships that allow inconsistent policies to coexist, together with tooling to analyse them. The relevant historical observation is modest and is all this paper claims from these three works: a literature that started with privilege ordering found, over roughly a decade, that ordering alone did not settle policy conflict, and built additional apparatus for the residue.

That history does not prove anything about language models. It does suggest that the residue is a normal feature of ordering-based designs rather than a sign that the ordering was drawn wrongly, and

it means the agent literature is re-encountering a problem with a substantial prior treatment it does not currently cite.

One disanalogy should be registered before the parallel is leaned on. In an operating system, the privilege level of a request is a property of the requesting process, known to the kernel with certainty. In an agent, the privilege level of an instruction is inferred from where the text appeared in a context window, and the inference can itself fail. Kariyappa and Suh (2026) report instruction identification as a distinct failure locus for exactly this reason. An operating system does not misread a ring number. This makes the agent case strictly harder than the systems case, and it means results imported from the older literature transfer as framing rather than as guarantees.

4. The case for the ordering paradigm, stated first

The instruction hierarchy works, and the evidence that it works has improved markedly in the last year. A reader who takes only Section 5 from this paper will have the balance wrong.

Wallace et al. (2024) reported that training GPT-3.5 on hierarchy-consistent data "drastically increases robustness -- even for attack types not seen during training -- while imposing minimal degradations on standard capabilities." Generalisation to unseen attacks is the property that matters for a defence, and it was present from the first paper.

Guo et al. (2026) sharpened this considerably. Fine-tuning GPT-5-Mini on an adversarially generated training set improved instruction-hierarchy robustness by 10.0 percentage points on average across 16 in-distribution, out-of-distribution and human red-teaming benchmarks, moving from 84.1 percent to 94.1 percent, while reducing unsafe behaviour from 6.6 percent to 0.7 percent and improving helpfulness on general safety evaluations. Those are large numbers on a broad evaluation suite, and the unsafe-behaviour reduction is close to an order of magnitude.

Zheng et al. (2026), published in Findings of ACL 2026, reframed hierarchy resolution as a reasoning task in which the model reasons about the relationship between a user prompt and higher-priority instructions before answering. They report roughly 20 percent absolute improvement in conflict setups across multiple model families, and reductions in attack success rate of up to 20 percentage points, including on safety-critical scenarios outside the training distribution.

Kariyappa and Suh (2026) report that training-free self-monitoring reduces rule-following non-compliance by 81 to 99 percent, with reductions of 86 percent under static attacks for one frontier model.

Two qualifications belong here rather than in a later section, because they bear on how much the numbers above settle.

The first is scope. Every result in this section is measured on conflicts that carry a privilege distinction. Wallace et al. (2024) trained on lower-versus higher-privilege instruction pairs; Zheng et al. (2026) constructed aligned and conflicting system-user pairs; Guo et al. (2026) targeted jailbreaks, system prompt extraction and agentic prompt injection. These are the cases the ordering is defined on. That the remedy performs well there is what one would expect if the analysis in this paper is right, and is not evidence against it.

The second is that the reduction under adaptive pressure is smaller than under static pressure. Kariyappa and Suh (2026) report 86 percent reduction under static attacks and 45 percent under adaptive attacks for the same model and monitor. That gap is a familiar pattern in security evaluation and is a reason to read static-benchmark gains as upper bounds.

5. Three failure classes under one name

The phrase "instruction hierarchy failure" currently covers at least three structurally different situations. They are separated here because they have different remedies, and because a benchmark that mixes them reports a number that no single intervention can move.

5.1 Class A: cross-tier conflict, ordering defined

Two instructions carry different privilege labels and disagree. A user turn asks the model to reveal the system prompt; a tool output instructs the agent to exfiltrate a file. The ordering states the answer, the training objective is well-posed, and the results in Section 4 apply. This class contains most of the prompt-injection literature.

Within Class A there is a genuine difficulty of scale that should not be mistaken for the problem this paper is about. Zhang et al. (2026) observe that the dominant paradigm assumes "a fixed, small set of privilege levels (typically fewer than five) defined by rigid role labels," and built ManyIH-Bench to test resolution across up to 12 levels, over 853 agentic tasks spanning 46 real-world agents. They report that even frontier models perform poorly, at roughly 40 percent accuracy, when instruction conflict scales. That is a hard problem, and it is still Class A: an ordering exists and the model fails to apply it. The remedy is better application of a defined rule.

5.2 Class B: same-tier conflict, ordering undefined

Two instructions carry the same privilege label and disagree. Here the ordering returns no answer, and the failure is not a failure to apply a rule but the absence of one.

The clearest measurement is Yan et al. (2026). Their profiler extracts rules from real prompt policies, encodes them, identifies pairs that can jointly govern the same pre-generation state, and realises those pairs as concrete test cases. From six public policies they extracted 276 source rules and 560 clauses, classified 30,944 clause-pair comparisons, retained 170 encoded hard-collision rule pairs, and realised 1,402 witnesses yielding 13,335 judgeable trials. Only 35.4 percent of those trials satisfied both governed rules. The framing in their own title is precise and worth preserving: these are within-policy collisions, where "individually reasonable standing rules can jointly govern the same pre-generation state."

He et al. (2025) approach the same structure from inside a single instruction. Their benchmark evaluates detection and resolution of conflicting constraints within user instructions. Two findings matter here. Detection is largely solved for strong models: they report F1 scores of 91.5 percent for DeepSeek-R1 and 87.3 percent for Claude-4.5-Sonnet. Resolution behaviour is not: models "rarely explicitly notify users about the conflicts or request clarification when faced with conflicting constraints." A model that reliably detects a conflict it has no rule to resolve, and then silently picks a side, is producing exactly the behaviour an undefined case predicts.

Yang et al. (2026a) supply the multi-principal version. They formalise multi-user agent interaction as a multi-principal decision problem in which one agent must serve several users with distinct roles, preferences and authority levels, under conflicting interests, information asymmetry and privacy constraints. Where two principals hold equal authority, no ordering over source types distinguishes them; the conflict is between two instances of one level.

A fourth instance of the structure is worth naming because it is so widely deployed that it is easy to overlook. Bai et al. (2022) train harmlessness from a written list of rules or principles, with the list supplying the only human oversight in the procedure. Their description of the method is a list of principles, not a ranking of them, and no privilege ordering over the principles is reported. Principles stated separately can bear on the same case in opposite directions: one favouring helpfulness and one favouring caution are each satisfiable alone and jointly unsatisfiable on some inputs. If the list carries no ranks, whatever resolves that tension during training is not a privilege ordering. On that reading,

constitutional methods already depend on some same-tier resolution behaviour, supplied implicitly by the preference model rather than stated as a rule. This is an inference from how the method is described rather than a reported property of it, and a ranking applied in implementation would defeat it.

5.3 Class C: neither, or not a conflict problem at all

Kariyappa and Suh (2026) provide the decomposition that makes this class visible. A non-compliant response, they observe, can arise because the model failed to identify the relevant instructions in context, or failed to resolve the conflict among instructions it did identify, or resolved the conflict correctly in its reasoning and still produced a violating response. Only the middle locus is a conflict-resolution failure. They report that the dominant failure mode varies across models, tasks and context length.

That last finding is load-bearing for this paper, and it cuts in a direction worth stating plainly: if the dominant locus varies by condition, then a single compliance score aggregated across conditions is not measuring a stable property. Guo et al. (2026) make a compatible observation from the training side, noting that hierarchy failures can be confounded with ordinary instruction-following failures, and that models can learn to overrefuse instead of learning the ordering. Overrefusal is a well-documented failure mode in its own right (Pan et al., 2025), and a training signal that cannot distinguish "correctly declined a lower-privilege instruction" from "declined too much" is one a model can satisfy the wrong way.

Potham (2025) supplies a second reason to distrust compliance scores as a measure of adjudication, from a benchmark that pits a high-level safety principle against conflicting task instructions across six models. Two findings are reported: a quantifiable cost of compliance, where safety constraints degrade task performance even when a compliant solution exists, and an illusion of compliance, where high adherence often masks task incompetence rather than principled choice. The second is the sharper one for this paper. A model that cannot do the task will not violate the constraint, and will score as compliant. Any compliance metric that does not condition on capability is therefore partially measuring inability, and the direction of that bias flatters exactly the models least able to create the conflict in the first place. This is a single-author preprint and the finding has not been replicated, but the mechanism it names does not depend on its measurements: it follows from what a compliance score counts.

Class C also collects failures that are specification-level rather than resolution-level. Acharya (2026) reports that enterprise multi-agent deployments exhibit failure rates between 41 and 86.7 percent, with nearly 79 percent of failures originating from specification and coordination issues rather than model capability limits. Cemri et al. (2025) assemble a dataset of more than 1,600 annotated traces across seven multi-agent frameworks and build a 14-mode failure taxonomy from a rigorous analysis of 150 of them, reporting inter-annotator agreement of kappa equal to 0.88 on that development analysis. Their three top-level categories are system design issues, inter-agent misalignment and task verification, which places a large share of failures away from any single adjudication step. Li et al. (2026) observe that in long-horizon agent workflows a system may produce a compliant final answer while leaking private information along the way, which makes end-of-episode compliance an incomplete measure.

6. Why the classes are not currently separated

The three classes are measured together.

Zhang et al. (2025) introduced IHEval, comprising 3,538 examples across nine tasks covering cases where instructions at different priorities either align or conflict, and reported that all evaluated models decline sharply under conflict, with the most competitive open-source model reaching only 48 percent accuracy in resolving conflicts. The design is explicitly organised around priority differences, which makes it a Class A instrument, and it does not claim otherwise. The difficulty arises when its headline

number is read as a general measure of how well models handle conflicting instructions.

McCauley et al. (2026) come closest to separating surfaces. Across 37 models they report hierarchy compliance ranging from 98.2 percent down to 20.5 percent, and find that strong system-over-user compliance is not a reliable proxy for user-over-tool robustness: several models preserve system constraints under direct user conflict but degrade sharply when the conflicting instruction arrives in a tool output. They also report that constraint hardening splits models, fixing some failures with stronger warnings while others persist at all strictness levels, and that the most revealing failures are subtle rather than overtly dangerous. Their conclusion, that hierarchy robustness is a set of behaviours rather than a single capability, is a Class A refinement: both surfaces they separate are cross-tier. The paper is nonetheless the strongest existing evidence that aggregate compliance scores conceal structure.

A 78-point spread across models on a single benchmark, and a demonstrated dissociation between two conflict surfaces within it, are hard to interpret if the underlying quantity is a sum over classes with different remedies. This is an inference about measurement, not a demonstrated confound; no source found during this search reports a same-tier versus cross-tier breakdown of any published hierarchy-compliance number. That breakdown does not appear to exist, and its absence is the specific gap this section names.

The instruction-following literature that predates the hierarchy framing had a cleaner separation available, because it evaluated verifiable constraints one at a time (Zhou et al., 2023). Huang et al. (2026) return to that discipline for coding agents, scoring operational rules individually from execution evidence, and note the reason it matters: when an agent obeys a rule, it may simply have been going to do that anyway. Per-rule scoring against execution evidence is the kind of instrument that could report Class A and Class B separately. None of the hierarchy benchmarks surveyed here does so.

There is a second, subtler reason the classes resist separation, which concerns what a benchmark can score. Scoring requires a key. For a cross-tier conflict a key is available by construction: the higher-privileged instruction is the correct answer, and compliance is well-defined. For a genuine same-tier conflict between two individually reasonable rules there is no such key, because the paradigm that generates keys does not reach the case. A benchmark author facing this has three options. They can decline to include such items, which is what most hierarchy benchmarks do, and which makes the reported number a Class A number by construction. They can impose a tiebreak of their own, which converts the item into Class A by fiat and measures conformity to the author's tiebreak rather than to anything about hierarchy. Or they can score both-satisfied as the only success, which is the choice Yan et al. (2026) make and which is why their figure is a satisfaction rate rather than a compliance rate.

That third option is why the 35.4 percent of Yan et al. (2026) and the 48 percent of Zhang et al. (2025) cannot be compared, and it is a sharper reason than the general incommensurability noted in Section 10. They are not two measurements of one quantity taken with different instruments; they are measurements of different quantities. One asks whether a model picked the privileged instruction, and the other asks whether it managed to satisfy two rules at once.

7. What a same-tier rule would have to supply

An ordering supplies three things at once, which is why its absence is felt as more than a missing tiebreak. It supplies a total order, so any two items are comparable. It supplies a justification, since the order encodes trust. And it supplies determinism, so the same conflict resolves the same way twice.

A same-tier resolution rule would have to supply substitutes for all three, and the substitutes are not obviously available.

Comparability. Two standing rules in one policy, or two users with equal authority, are equally

privileged by construction. Any rule that makes them comparable must appeal to something other than privilege: recency, specificity, scope of harm, or an explicit meta-policy written by the policy author. Each of these is a design decision with consequences, and none is implied by the hierarchy.

Justification. Cross-tier resolution is justified by trust: the system prompt wins because the platform is more trusted than the user. No analogous justification is available within a tier. A recency rule, for instance, would mean a later-written standing rule silently overrides an earlier one, which is a policy-authoring hazard rather than a safety property.

Determinism, and the option of declining to resolve. This is where the literature already contains a partial answer and a partial disagreement. He et al. (2025) show that strong models detect intra-instruction conflicts reliably while rarely surfacing them, which means the capability needed to escalate rather than silently resolve is largely present and is not being used. Escalation is a legitimate resolution for a genuinely undefined case, and it is not scored by any hierarchy benchmark surveyed here.

Yang et al. (2026b) offer the most developed mechanism found during this search. Their approach formulates instruction resolution as a constraint satisfaction problem and derives a maximally consistent set of applicable instructions under hierarchical constraints, with solver decisions distilled into model parameters. This is a real advance on taking a maximum, and it directly targets partial and implicit conflicts. Two things should be said about it. It is the strongest counter to this paper's framing, because a maximally consistent set is defined even when the ordering is partial. But constraint satisfaction returns a maximally consistent set, not a uniquely correct one, and when two same-tier rules are individually satisfiable and jointly not, more than one maximal set exists. Something outside the solver must choose. Whether that residual choice is small or large in practice is not reported, and appears not to have been measured.

The delegation literature is developing a parallel apparatus for authority in multi-agent settings, including delegation contracts with attested identity (Prakash, 2026), verifiable delegation chains with authority narrowing (Patil, 2026), and runtime delegation safety frameworks (Sun, 2026). These address who may authorise what, which is a cross-tier question, and none of them supplies a within-tier tiebreak. They are noted here because they are sometimes read as covering the same ground.

Three operational consequences follow if the analysis above is right, and they are stated as conditionals because none has been measured.

An aggregate compliance score is the wrong instrument for a deployment decision. A team choosing between two models on a hierarchy benchmark is comparing sums over classes whose mix in their own traffic is unknown. McCauley et al. (2026) already demonstrate the weaker version of this within Class A, reporting that strong system-over-user compliance does not predict user-over-tool robustness. If the same non-transfer holds between classes, a model selected for Class A performance carries no warranty on Class B, and nothing currently reports that separately.

Silent resolution is the failure mode to watch for, not refusal. A model facing an undefined conflict does not announce that the case is undefined. He et al. (2025) report that models detect intra-instruction conflicts at F1 above 0.87 for strong models while rarely surfacing them, which describes a system that knows something the operator is not told. In a Class A failure the agent does something it was forbidden to do, which is legible in an audit log. In a Class B failure the agent satisfies one legitimate rule and quietly drops another, which is not obviously distinguishable from correct behaviour after the fact.

Policy growth is the mechanism that would make this worse over time. Yan et al. (2026) extract 276 rules from six policies and find 170 hard-collision rule pairs among 30,944 clause-pair comparisons. Pairwise collision opportunities grow quadratically in the number of standing rules, while the privilege structure over sources does not grow at all. A team that responds to each incident by appending a rule

to the system prompt is, on this account, adding Class B surface at a rate faster than linear. Whether observed collision counts actually follow that growth curve is not established by any source found here, and the second study named in Section 8 is the measurement that would show it.

8. Studies that would settle the open parts

Each item names a measurement that does not appear in any source found during this search.

The first is the cheapest and would settle the most. Re-score IHEval, ManyIH-Bench or IH-Benchmark with each item labelled by whether the conflicting instructions carry distinct privilege labels. This is annotation, not new data collection, and it would show directly how much of each reported compliance number is Class A.

The second is a prevalence study. The ratio of same-tier to cross-tier instruction conflict in production agent logs is unknown. Without it, the practical weight of the argument made here cannot be assessed, and this paper does not assert what that ratio is.

The third tests transfer. Take a model trained on an instruction-hierarchy dataset and evaluate it on within-policy collisions of the kind Yan et al. (2026) realise. If ordering-based training helps on conflicts the ordering does not define, the analysis in this paper is wrong in an interesting way.

The fourth tests interference. Hold a cross-tier conflict fixed and add an unrelated same-tier collision to the same policy. Section 6 predicts that the added collision degrades cross-tier compliance; nothing currently measures it.

The fifth makes escalation scoreable. Allow declining to resolve, and asking, to count as a correct answer for genuinely undefined conflicts, then measure how often models choose it when it is available and rewarded. He et al. (2025) suggest the underlying detection capability is already present.

The sixth quantifies the residue left by constraint satisfaction. For the method of Yang et al. (2026b), report how often more than one maximally consistent instruction set exists, and how the implementation chooses between them.

The seventh quantifies overrefusal as a confound. Guo et al. (2026) name the shortcut. Reporting hierarchy gains alongside a matched overrefusal measure on benign same-tier conflicts would show how much of a reported gain is discrimination and how much is caution.

The eighth establishes whether the failure locus is stable. Kariyappa and Suh (2026) report that the dominant locus varies by condition. Measuring that variation across a systematic grid of models, tasks and context lengths would establish whether a single compliance score can be a stable construct.

The ninth is a replication. The 35.4 percent figure rests on one profiler over six policies. It is the single most load-bearing number in this paper and has not been replicated.

9. The case against this paper's position

The strongest objections are stated here in the form a reviewer would use.

The distinction may be a distinction without a difference. If, in practice, policy authors resolve same-tier collisions by editing the policy, then the problem is a policy-authoring problem and not a model problem, and the ordering paradigm is not obliged to solve it. Yan et al. (2026) profile policies as written, not policies as maintained. A team that runs a conflict analyser over its prompt policy before deployment removes most of Class B without any model change. This objection is strong and this paper cannot answer it with evidence.

Class B may reduce to Class A under a finer labelling. Two standing rules in one system prompt

carry the same label only because the labelling is coarse. A policy author could annotate rules with explicit priorities, converting the conflict to cross-tier. Zhang et al. (2026) push in exactly this direction with many-tier hierarchies, and Yang et al. (2026b) with hierarchical constraints over partially ordered instructions. If arbitrary tiering is always available, "undefined" becomes "not yet specified," which is a weaker claim than the one this paper makes. The counter-argument is that two equally authorised users cannot be tiered without deciding which user matters more, which is a governance decision rather than a labelling one. That counter-argument covers the multi-principal case and does not obviously cover the within-policy case.

The empirical base for Class B is thin. The central quantitative claim rests substantially on one paper (Yan et al., 2026). It is an unreplicated preprint with three authors, and its 35.4 percent figure comes from a pipeline with many design choices: which rules are extracted, which pairs count as hard collisions, and how witnesses are realised. A different pipeline could yield a materially different number. Section 5.2 would survive its loss, because He et al. (2025) and Yang et al. (2026a) establish the structure independently, but the quantitative force of the section would not.

The paradigm is already moving. Yang et al. (2026b) explicitly address benign, partial and implicit conflicts; McCauley et al. (2026) already argue that hierarchy robustness is not a single capability; Kariyappa and Suh (2026) already decompose the failure. A reviewer could reasonably say that this paper names a direction the field entered without it, and that the partition offered here is a relabelling of observations already in print. That is the most serious threat to the contribution, and it is not fully answerable. What is offered in response is narrow: none of those three papers distinguishes conflicts by whether an ordering is defined over them, and that specific cut is what generates the studies in Section 8.

Ordering may be sufficient in the limit. If instruction-hierarchy training continues to improve at the rate between Wallace et al. (2024) and Guo et al. (2026), residual same-tier failures may become small enough not to matter. This paper's reply is that a rate of improvement on Class A does not bound Class B error, because the training signal for Class B is not defined. But that reply is an argument, not a measurement.

Escalation may be a worse remedy than silent resolution. Section 7 treats asking the operator as a legitimate outcome for an undefined case. An agent that escalates every same-tier collision in a 276-rule policy would be unusable, and the overrefusal literature (Pan et al., 2025) documents what happens when caution is trained without discrimination: models decline benign requests. Guo et al. (2026) name overrefusal specifically as a shortcut models learn when the hierarchy objective is hard. It is therefore possible that silent resolution is the pragmatic optimum and that the correct response to Class B is better policy authoring rather than any model behaviour at all. This paper does not have an argument against that position, only the observation that the choice is currently being made implicitly by systems that do not report making it.

The three classes may not be exhaustive or exclusive. The partition is offered as an analytic aid, not a theorem. A single failure can plausibly involve an identification error and a same-tier collision at once, and Class C is defined partly by exclusion, which is a sign it is doing residual work rather than carving anything. A reviewer who regards Class C as a bucket rather than a category is reading it correctly.

10. Threats to the argument, and the quality of the evidence

Source quality is mixed and is stated rather than implied. Of the load-bearing sources, three are refereed: Zhang et al. (2025) at NAACL 2025, Zheng et al. (2026) in Findings of ACL 2026, and Andriushchenko et al. (2025) at ICLR 2025. The remainder are unreplicated preprints. Five of the works

cited, in Sections 5.3 and 7, are single-author preprints (Potham, 2025; Acharya, 2026; Prakash, 2026; Patil, 2026; Sun, 2026). No claim in this paper rests on any of them alone: they are cited for framing and for the existence of proposals, not for the argument's load-bearing quantities. The one place a single-author preprint carries argumentative weight is the illusion-of-compliance point in Section 5.3, and the text there states that the mechanism follows from what a compliance score counts rather than from that paper's measurements.

The load-bearing quantities are: the 35.4 percent within-policy figure (Yan et al., 2026, three authors, unrepliated); the detection-resolution dissociation (He et al., 2025, seven authors, unrepliated); the three-way failure decomposition and its condition-dependence (Kariyappa and Suh, 2026, two authors, unrepliated); and the training gains in Section 4 (Guo et al., 2026, thirteen authors, unrepliated; Zheng et al., 2026, refereed). Section 5.2 falls if Yan et al. (2026) does not survive scrutiny, and Section 6's inference is weakened but not removed if Kariyappa and Suh (2026) does not.

Three further limitations belong on the record.

The argument is structural, and structural arguments are cheap. That an ordering does not induce an order within a level is true by definition and therefore carries no empirical content by itself. All of the empirical content is in the claim that the undefined case occurs often enough to matter, and that claim rests on the sources named above.

Numbers from different papers are not commensurable. The 35.4 percent of Yan et al. (2026), the 48 percent of Zhang et al. (2025), the roughly 40 percent of Zhang et al. (2026) and the 20.5-to-98.2 percent range of McCauley et al. (2026) are computed over different task populations with different scoring protocols, and are quoted here as individual findings, never as a series.

Finally, this paper takes each cited design at its authors' description. Where an abstract reports a design, that description has been used as written; full-text verification of every design choice in every cited work was not performed, and a mischaracterisation of a design would propagate into the partition offered here.

References

- Acharya, V. (2026). Semantic Consensus: Process-Aware Conflict Detection and Resolution for Enterprise Multi-Agent LLM Systems. arXiv:2604.16339. [PREPRINT]
- Andriushchenko, M., Souly, A., Dziemian, M., Duenas, D., Lin, M., Wang, J., Hendrycks, D., Zou, A., Kolter, Z., Fredrikson, M., Winsor, E., & Wynne, J. (2025). AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. International Conference on Learning Representations (ICLR) 2025. arXiv:2410.09024.
- Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., & Olsson, C. (2022). Constitutional AI: Harmlessness from AI Feedback. arXiv:2212.08073. [PREPRINT]
- Cemri, M., Pan, M. Z., Yang, S., Agrawal, L. A., Chopra, B., Tiwari, R., Keutzer, K., Parameswaran, A., Klein, D., Ramchandran, K., Zaharia, M., & Gonzalez, J. E. (2025). Why Do Multi-Agent LLM Systems Fail? arXiv:2503.13657. [PREPRINT]
- Debenedetti, E., Zhang, J., Balunovic, M., Beurer-Kellner, L., Fischer, M., & Tramer, F. (2024). AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. arXiv:2406.13352. [PREPRINT]
- Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. arXiv:2302.12173. [PREPRINT]
- Guo, C., Ceron Uribe, J. F., Zhu, S., Choquette-Choo, C. A., Lin, S., Kandpal, N., Nasr, M., Toyer, S., Wang, M., Yu, Y., & Beutel, A. (2026). IH-Challenge: A Training Dataset to Improve Instruction Hierarchy on Frontier LLMs. arXiv:2603.10521. [PREPRINT]
- He, X., Zhang, Q., Chen, P., Chen, G., Yu, L., Yuan, Y., & Yiu, S.-M. (2025). ConInstruct: Evaluating Large Language Models on Conflict Detection and Resolution in Instructions. arXiv:2511.14342. [PREPRINT]

- Huang, Z., Que, H., Zeng, H., Zhang, G., Wang, Z., Chen, J., Wang, H., Hou, Z., Pu, C., Yan, S., & Huang, W. (2026). Harness-IF: Evaluating Instruction Following Across Instruction Surfaces in Coding Agents. arXiv:2608.11727. [PREPRINT]
- Kariyappa, S., & Suh, G. E. (2026). Where Instruction Hierarchy Breaks: Diagnosing and Repairing Failures in Reasoning Language Models. arXiv:2606.07808. [PREPRINT]
- Li, T., Ma, Y., Wen, H., Huang, Z., Zhou, Q., Fu, Z., & Cheng, G. (2026). Safe Multi-Agent Behavior Must Be Maintained, Not Merely Asserted: Constraint Drift in LLM-Based Multi-Agent Systems. arXiv:2605.10481. [PREPRINT]
- Lupu, E. C., & Sloman, M. (1999). Conflicts in policy-based distributed systems management. *IEEE Transactions on Software Engineering*, 25(6), 852-869. doi:10.1109/32.824414
- McCauley, C., Kan, Z., & Martin, J. (2026). IH-Benchmark: A Conflict-Centered Benchmark for Instruction-Hierarchy Robustness in LLM Applications. arXiv:2607.25987. [PREPRINT]
- Moffett, J. D., & Sloman, M. S. (1993). Policy hierarchies for distributed systems management. *IEEE Journal on Selected Areas in Communications*, 11(9), 1404-1414. doi:10.1109/49.257932
- Moffett, J. D., & Sloman, M. S. (1994). Policy conflict analysis in distributed system management. *Journal of Organizational Computing*, 4(1), 1-22. doi:10.1080/10919399409540214
- Pan, L., Tong, Y., Zhang, X., Zhang, X., Zhou, J., & Chu, Z. (2025). Understanding and Mitigating Overrefusal in LLMs from an Unveiling Perspective of Safety Decision Boundary. arXiv:2505.18325. [PREPRINT]
- Patil, K. (2026). SentinelAgent: Intent-Verified Delegation Chains for Securing Federal Multi-Agent AI Systems. arXiv:2604.02767. [PREPRINT]
- Perez, F., & Ribeiro, I. (2022). Ignore Previous Prompt: Attack Techniques For Language Models. *ML Safety Workshop, Conference on Neural Information Processing Systems (NeurIPS) 2022*. arXiv:2211.09527.
- Potham, R. (2025). Evaluating LLM Agent Adherence to Hierarchical Safety Principles: A Lightweight Benchmark for Probing Foundational Controllability Components. arXiv:2506.02357. [PREPRINT]
- Prakash, S. (2026). The Provenance Paradox in Multi-Agent LLM Routing: Delegation Contracts and Attested Identity in LDP. arXiv:2603.18043. [PREPRINT]
- Saltzer, J. H., & Schroeder, M. D. (1975). The protection of information in computer systems. *Proceedings of the IEEE*, 63(9), 1278-1308. doi:10.1109/proc.1975.9939
- Schroeder, M. D., & Saltzer, J. H. (1972). A hardware architecture for implementing protection rings. *ACM SIGOPS Operating Systems Review*, 6(1). doi:10.1145/850614.850623
- Sun, Y. (2026). Safe Bilevel Delegation (SBD): A Formal Framework for Runtime Delegation Safety in Multi-Agent Systems. arXiv:2604.27358. [PREPRINT]
- Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., & Beutel, A. (2024). The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. arXiv:2404.13208. [PREPRINT]
- Yan, L., Chen, X., & Zhang, X. (2026). WIRE: Profiling Witnessed Within-Policy Instruction Collisions in LLM Agents. arXiv:2605.27784. [PREPRINT]
- Yang, S., Zhu, S., Zhu, H., Enriquez, J. R., Wang, D., Pentland, A., Bakker, M. A., & Pei, J. (2026a). Multi-User Large Language Model Agents. arXiv:2604.08567. [PREPRINT]
- Yang, S., Zhou, Z., Wang, D., & Li, W. (2026b). Hierarchical Alignment: Enforcing Hierarchical Instruction-Following in LLMs through Logical Consistency. arXiv:2604.09075. [PREPRINT]
- Zhang, J., Li, T., Jurayj, W., Zhan, H., Van Durme, B., & Khashabi, D. (2026). Many-Tier Instruction Hierarchy in LLM Agents. arXiv:2604.09443. [PREPRINT]
- Zhang, Z., Li, S., Zhang, Z., Liu, X., Jiang, H., Tang, X., Gao, Y., Li, Z., Wang, H., Tan, Z., Li, Y., & Yin, Q. (2025). IHEval: Evaluating Language Models on Following the Instruction Hierarchy. *North American Chapter of the Association for Computational Linguistics (NAACL) 2025*. arXiv:2502.08745.
- Zheng, Z., Balachandran, V., Park, C. Y., Brahman, F., & Kumar, S. (2026). Reasoning Up the Instruction Ladder for Controllable Language Models. *Findings of the Association for Computational Linguistics: ACL 2026*, 39332-39354. arXiv:2511.04694.
- Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., & Hou, L. (2023). Instruction-Following Evaluation for Large Language Models. arXiv:2311.07911. [PREPRINT]

(c) 2026 Pranay Mahendrakar. Licensed under CC BY 4.0.

The literature search, drafting and citation verification for this paper were carried out with AI assistance under the author's direction. Every citation was machine-verified against the arXiv API and Crossref before inclusion, and every quantitative claim was read back against the cited source's own abstract. The author is responsible for the final text and for all claims made in it.